インターネットにおける可用性改善のための代替サーバ利用技術

門林雄基[†] 山口英[††] 宮原秀夫[†††]

TCP/IPプロトコルに基づくインターネットワークではトランスポート層における可用性が低く，観測に基づく結果によれば約90%の可用性しか達成していない．大部分のアプリケーションはトランスポート層を直接利用しているため，アプリケーションがユーザに対して提供するサービスの可用性も著しく低い．本研究では，このような環境における情報アクセスの可用性を改善するための方式として，サーバ・クライアント間へのアクセス装置の導入を提案し，アクセス装置における代替サーバ利用技術の実現方式について述べる．本方式に基づいてアクセス装置csdを実装し，そのインターネットにおける運用を通じて可用性の改善効果を確認した．

Techniques for Using Alternative Servers to Improve Service Availability in the Internet

YOUKI KADOBAYASHI,[†] SUGURU YAMAGUCHI[††] and HIDEO MIYAHARA[†††]

Many users of Internet services have experienced some kind of network troubles, which keep them from sending e-mail or accessing web servers. Such troubles have been happening frequently, since most services are built directly on top of transport layer, and since transport layer does not achieve high availability; according to our measurements, its availability was approximately 90%. We propose a strategy for improving service availability in Internet, which consists of 1) transparent introduction of access engine between clients and servers, and 2) implementation techniques for utilizing alternative servers at the access engine. We implemented our access engine, called csd, based on this technique. We confirmed its effectiveness through the deployment of csd in the Internet.

1. はじめに

今日のインターネットはゲートウェイモデルに基づいて設計，実装されている[1),2)]．ゲートウェイモデルでは，ホストとルータに実装されたプロトコルスタックの各層に，異なった視点から設計された信頼性実現のための機能が組み込まれており，プロトコルスタック全体としてサービスの信頼性を提供する．このモデルに基づいて設計，実装されたプロトコルスタックの1つが，現在のインターネットの基盤であるTCP/IPプロトコルスタックである．しかしながら，現在のTCP/IPプロトコルスタックは終端間のデータストリームの信頼性しか提供しておらず，終端間のサービスの可用性（availability）を保証する機構は実現されていない．

一方，インターネットにおける観測によれば，主要なトランスポート層プロトコルであるTCPにおいてコネクション確立に失敗する確率は約10%ときわめて高い．現在のインターネット上のアプリケーションの大部分はトランスポート層が提供する信頼性機能だけに頼って実装されている．このため，トランスポート層以下において障害が起きた場合，サーバに接続できなくなる等の不具合が起き，目的とする作業を遂行することができなくなる可能性が高い．具体的には，必要とする情報にアクセスすることができなくなる，あるいは電子メールを配送することができなくなる等の問題が生じる．

このように，現在のインターネットではサービスの可用性が著しく低いため，社会における情報通信基盤としては不十分であると考えられる．インターネット

† 大阪大学大型計算機センター
Computation Center, Osaka University
†† 奈良先端科学技術大学院大学情報科学研究科
Graduate School of Information Science, Nara Institute of Science and Technology
††† 大阪大学基礎工学部情報工学科
Faculty of Engineering Science, Osaka University

が次世代における情報通信基盤となるためには，サービスの可用性を保証するための機構が必要とされている．また，そのような機構はできるだけ多くのアプリケーションに対して効果をもたらすことが望ましい．

このためのアプローチとしては様々な手法が考えられる．前論文[3]では，既存のクライアント・サーバ型アプリケーションに対して透過的にサービスの可用性を付加することができるアクセス装置方式を提案した．提案方式では，地理的に離れた複数のサーバを相互にバックアップする運用形態を前提とし，クライアント・サーバ間に位置するアクセス装置においてサーバ切替えを行う．

本研究では，アクセス装置における代替サーバ利用技術の実現方式を提案する．ボリュームを基本とした代替サーバの構成手法と，アクセス装置における代替サーバ利用技術を確立することで，可用性を改善することができると期待される．提案方式に基づき，サーバ切替え機能を分散ファイルシステム上のアクセス装置 csd に実装した．csd のインターネット上での運用を通じて得られたデータに基づき，アクセス装置を用いることによる可用性の改善効果を確認することができた．

1.1 インターネットの障害特性

TCP は TCP/IP プロトコルスタックの中心となるトランスポート層プロトコルであり，TELNET（遠隔ログイン），SMTP（電子メール配送）を始めとする多くのアプリケーション層プロトコルの下位層として用いられている．したがって，ネットワークアプリケーションがつねに安定して動くと仮定した場合，インターネットの障害特性は TCP の障害特性とほぼ同じであると考えることができる．言い換えれば，ユーザからみたサービスの可用性は TCP コネクションの信頼性とほぼ同じである．

実際の運用ネットワークでは高い確率で TCP コネクションの確立に失敗したり，TCP コネクションが通信途中で切断される．インターネットにおけるトランスポート層プロトコルの信頼性について具体的な知見を得るため，奈良先端科学技術大学院大学，電気通信大学，東京大学などにおいて TCP のエラー状況を測定した．測定時の負荷として TELNET，FTP 等を選んだ場合は接続先が限定されるため，ここではインターネット上のさまざまな計算機に接続するアプリケーションとして WWW を選び，TCP コネクションが集中する WWW 代理サーバ（proxy server）上で netstat -s を用いて測定を行った（表 1）．その結果，インターネットにおいて TCP コネクションが切

表 1 TCP コネクションにおける障害発生率
Table 1 TCP connection failure rates.

組織名	障害発生率（%）	期間（日）	総観測数
奈良先端大	4.9	52	1,817,927
電通大	8.6	11	558,071
東大	19.1	86	28,488
企業 1	5.6	27	228,282
企業 2	14.6	141	108,879

```
p_t = (tcps_drops + tcps_conndrops
      + tcps_timeoutdrop + tcps_keepdrops)
      / tcps_closed
```

図 1 p_t の計算式
Fig. 1 The formula for calculating p_t.

表 2 図 1 における各変数の意味
Table 2 Meanings of each variable in Fig. 1.

変数名	内容
tcps_drops	SYN 受信後に切断した本数
tcps_conndrops	SYN 受信前に切断した本数
tcps_timeoutdrop	再送タイムアウトで切断した本数
tcps_keepdrops	Keepalive で切断した本数
tcps_closed	閉じた本数（切断を含む）

断されるか，相手が接続要求に対して応答しない確率 p_t は 5%から 20%程度であるということが分かった．p_t は切断されたコネクションの総数を，閉じたコネクションの本数（切断を含む）で割ることで計算できる．p_t の計算式を図 1 に示す．これらの変数の値はオペレーティングシステム内部の tcpstat 構造体から得ることができる[4]．各変数の意味を表 2 に示す．

このように，インターネットでは TCP コネクションが切断されたりコネクション確立に失敗する確率が約 5%から 20%ときわめて高い．このことは，以下に述べるようなインターネットの構成方法に起因するものである．インターネットは多くのルータからなり，また，それらのルータのいずれかが故障した場合，代替経路におけるすべてのルータが正常に機能していなければネットワークが切断されてしまう．また，経路制御プロトコルは経路情報がすべて正しいという仮定のもとに設計されており，誤った経路情報が流された場合に誤動作する可能性がある．このため，ルータにおける人為的な操作ミスや，経路制御ソフトウェアのバグによって誤った経路情報が流され，特定のネットワークに対して到達できなくなってしまう場合がある．このように，ネットワーク層において到達性が保証されていないため，主要なトランスポート層プロトコルである TCP においてもコネクションの確立を保証することができない．また，接続先の計算機が応答しな

い場合や，接続先の計算機が再起動した場合においても過渡的な障害が起きる．

2. 代替サーバの利用技術

前章で述べたように，現在のインターネットにおけるトランスポート層プロトコルは十分な可用性を提供していない．大多数のネットワークアプリケーションはトランスポート層プロトコル上に直接実装されているため，ユーザに対して提供しているサービスの可用性が低い．このような状況においてサービスの可用性を改善するため，前論文[3]においてアクセス装置を提案した．

提案方式では，地理的に離れた複数のサーバを相互にバックアップする運用形態を前提としている．このような代替サーバが利用できなかった場合は可用性の改善効果は得られない．また，代替サーバの内容をつねに最新の状態に保っておくことが必要不可欠である．インターネットでは情報アクセスサービスにおいてこのような運用形態が一般的であることから，本研究では可用性向上の対象を情報アクセスに限定している．

2.1 アクセス装置による代替サーバ利用

アクセス装置はクライアントとサーバの間に介在し，2者間の通信をアプリケーション層において中継する．アクセス装置は各々のサーバの障害状態を把握し，アクセス要求を中継するときに障害が起きていないサーバを選択することでクライアントからみたサーバの可用性を改善する．

サーバ近傍のネットワークの障害や，サーバ自身の障害である場合，アクセス装置において代替サーバへの切替えを行うことで高い可用性を実現することができる．クライアント側において障害が起き，クライアントとアクセス装置が通信不能になった場合は可用性の改善効果は得られない．しかしながら，アクセス装置をクライアントの近傍に置くことで，クライアントとアクセス装置が通信不能になる確率はかなり小さくなり，トランスポート層の障害をクライアントから隠蔽することができると考えられる．しかしながら，障害の影響を最小限におさえるためには，以下の要件を満たす必要がある．

(1) サーバ切替え時におけるユーザの利便性を損なわないよう，代替サーバと元のサーバで同一内容を提供すること．
(2) 障害を早期に検出し，サーバが利用できない時間を最小化すること．
(3) サーバとの通信途中において検知した障害情報を他のコネクションと共有し，初期サーバ選択時に障害のあるサーバを回避すること．
(4) サーバが障害から回復した場合，そのことを早期に検出し，障害が頻発するような状況においても代替サーバが利用できること．

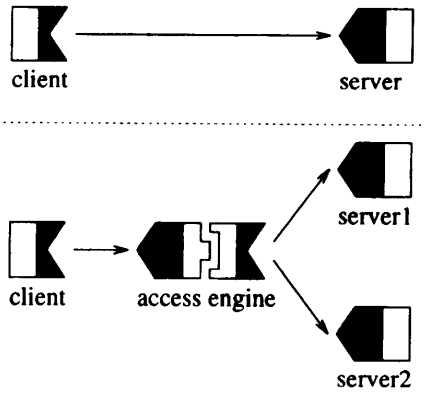


図2 アクセス装置の透過的な導入
Fig. 2 Transparent introduction of access engine.

従来の通信ソフトウェア実現方式ではこれらの要件を満たすことができない．代替サーバを効果的に利用するためには，新たな通信ソフトウェア実現方式に基づいてアクセス装置を実装する必要がある．その詳細については3，4章で述べる．

2.2 アクセス装置の透過的な導入

アクセス装置のサービスインタフェースをサーバのサービスインタフェースと同一にすることで，透過的にアクセス装置を導入することができ，既存のネットワークアプリケーションを変更することなく可用性の改善効果が得られる（図2）．たとえば，本研究で実装したアクセス装置 csd ではインターネット上のファイルに対するアクセスインタフェースとして従来のファイルシステムと互換性を持つものを採用し，アクセス要求を中継するときに代替サーバの選択を行う．サーバ切替え機能が csd 内部で実現されているため，FTP サーバなどファイルシステム上で動作する既存のサービスに対して透過的にサーバ切替え機能を付加することができる．

2.3 代替サーバ情報の記述

代替サーバは，インターネットのメール配送システム sendmail[5]において広く用いられている．これは，ネームサーバ[6]が持つ分散データベース内に各々のドメインに対する代替メールサーバの計算機名と代替サーバの優先順位を記述するというものである．たとえば図3のレコードは，iij.ad.jp ドメインに対してメールサーバが2台用意されており，ns.iij.ad.jp が ns1.iij.ad.jp より優先されるということを表している．

このようなアプローチを本研究が対象とする情報ア

```
iij.ad.jp.  MX  10   ns.iij.ad.jp.
iij.ad.jp.  MX  100  ns1.iij.ad.jp.
```

図3 sendmailにおける代替サーバの利用
Fig. 3 The use of alternative servers in sendmail.

クセスに適用することを考える．現在のインターネットにおける情報サービスは，多くの場合，あるまとまりを持った情報を単一のファイルとして提供している．また，インターネット全体からの平均アクセス時間の短縮などを目的として，地理的に離れた場所にファイルを複製するといったことがよく行われる．このような複製ファイルに対して代替サーバ手法を適用することを考える．ファイルごとに代替ファイルサーバの情報を記述した場合，ネームサーバにおいて必要となる記憶容量が爆発的に増大してしまうため，現実的でない．したがって，ファイルアクセスにおいて代替サーバを利用するためには，ファイルの集合に対して1つの代替ファイルサーバを記述するなど，何らかの方法を用いて代替ファイルサーバに関する記述を簡略化する必要がある．

2.4 ボリューム

以上のような背景から，本研究では大規模なインターネットワークにおいて複製ファイルを扱う単位としてボリュームを用いる．ボリュームとはファイルシステムの一部分を複製し，インターネット上に分散配置したものである．アクセス制御情報や代替サーバの情報などはファイルごとに記述することはせず，ボリュームを単位として記述する．ボリュームを単位としてサーバ切替えを行うため，代替サーバを簡潔に指定することができ，ネームサーバにおいて必要な記憶容量を少なくおさえることができる．

3. 従来の通信ソフトウェア実現方式における問題点

従来の通信ソフトウェア実現方式では，1つのプロセス内にサーバに対するコネクション1本を受け持つ単一のスレッドがあり，タイムアウトに基づいて障害検出を行う．また，多くの実装では障害発生時に一度プロセスを終了し，ユーザに対してサーバ名の再入力を求めるものが多い．

このような通信ソフトウェア実現方式をとった場合，以下のような問題点がある．トランスポート層で障害が起きたときに，ユーザが明示的にサーバを切り替えなければならない．障害が発生してからタイムアウトするまでの時間が長いため，利便性に問題がある．同一のサーバを利用している他のプロセスや他のクライアント上のプロセスがさきに障害を検知した場合においても，障害情報を共有することができない．また，ユーザの再試行に頼らずに障害からの回復を検出することができない．これらのことから，従来の通信ソフトウェア実現方式は2.1節で述べた要件を満たしていないことが分かる．

アクセス装置において代替サーバを利用するためには，新たな通信ソフトウェア実現方式を確立し，これらの問題点を解決する必要がある．

4. 代替サーバ利用技術の実現方式

本研究で実装したアクセス装置csdでは，代替サーバを効果的に利用するため，従来とは異なる通信ソフトウェア実現方式をとった．本方式によって2.1節で述べた要件を満たすことができると考えられる．

4.1 イベント駆動

アクセス装置はアプリケーション層においてクライアント・サーバ間のやりとりを中継するゲートウェイであるから，両者が用いているアプリケーション層プロトコルがTCP上に実装されている場合はTCPの処理と並行して障害を検出する必要がある．障害を早期に検出し，サーバが利用できない時間を最小化するためには，従来の通信ソフトウェアにおいてプロセス全体を待ち状態としていたTCPのconnect，accept処理を非同期で行う必要がある．このため，csdではコネクションごとに1つのスレッドを用意し，パケット，コネクション，シグナル，タイマ，スレッドスケジューリングをまとめて処理するイベントループを作成した．このような構造をとることで，コネクションの受付と並行して障害の検出を行うことができる．

4.2 ICMPを用いた障害検出

インターネットにおける障害は，ネットワーク層以下における障害とアプリケーション層における障害の2つに分けられる．さらに，それらの障害は過渡的な障害と長期的な障害に分類することができる．

ネットワーク層以下において障害が起きた場合，障害が起きたルータやホストの手前のルータが始点に対してICMP到達不可能メッセージを送ることがRFC1812において義務づけられている．通常のアプリケーションはこのメッセージを無視するが，アクセス装置ではこれを利用することで，サーバにいたる経路上で起きた障害を早期に検出し，代替サーバへの切替えを行うことができる．

アクセス装置では，障害発生から一定時間経過したのちに障害が起きたサーバに対してpingと同様の到達性確認処理を行うことで，過渡的な障害と長期的な障害を区別することができる．pingに対する応答があっ

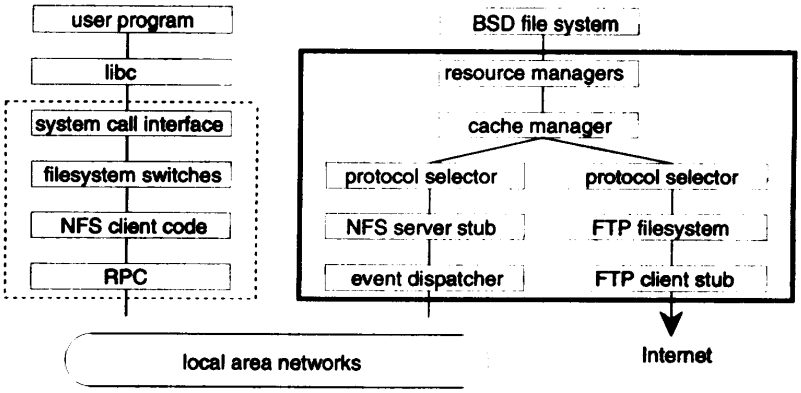


図4 アクセス装置 csd の構成（太線内）
Fig. 4 Structure of the csd access engine.

た場合は過渡的な障害であったと判断し，代替サーバから元のサーバに切り替える．応答がない場合は長期的な障害と判断する．

4.3 タイマを用いた障害検出

サーバのバグなどによってアプリケーション層において障害が起きた場合，アクセス装置ではアクセス要求を起点としたタイムアウトに基づく障害検出を行う．具体的には，すべてのコネクションについて，1分ごとにサーバからの新たな応答があったかどうかを調べ，2分以上応答がない場合はコネクションまたはサーバ上のプロセスに障害が起きたと判断してコネクションを閉じる．このとき同時に，到達性確認処理を行う．新たなアクセス要求が発生し，新たなコネクションが必要となった場合，到達性が確認できている場合にはサーバを切り替えずにコネクションの確立を試みる．到達性が確認できなかった場合にはサーバを切り替える．

このように，TCP のコネクションタイムアウト（約15分）に頼らずに障害を早期に検出することで，サーバを利用することができない時間を最小化することができると考えられる．

4.4 障害情報の共有

上で述べたような手法を用いて得られた障害情報をサーバおよびコネクションごとに管理することで，サーバとの通信途中において検知した障害情報を他のコネクションと共有することができ，また，初期サーバ選択時に障害のあるサーバを回避することができる．

5. アクセス装置の実装

我々はアクセス装置 csd を C 言語で実装し，BSD/OS, OSF/1, SunOS, Sony NEWS-OS, DEC Ultrix などの BSD を基本としたオペレーティングシステム上で動作確認した．ソースコードは約14,000行であり，資源管理部（ファイル管理，タイマ管理，スレッド管理，コネクション管理，サーバ管理），キャッシュファイルシステム，NFS プロトコルモジュール，FTP モジュール，ICMP モジュール，デバッグ用モジュールなどからなる．ユーザプログラムからのアクセス要求は NFS プロトコルによって伝えられ，csd においてキャッシュやサーバ選択などの処理が行われたのち，インターネット上のサーバへ FTP プロトコルを用いて伝えられる（図4）．それぞれのプロトコルモジュールはスレッドとして実装されている．スレッドの実現方式としては continuation[7] に基づくものを採用しているため，プラットフォームごとにアセンブリコードを書くことなく高いパフォーマンスが得られ，移植性との両立を実現している[8]．

ボリューム情報を記述する方式としては，DNS データベース上に記述する方式とファイル上に記述しファイルを配布する方式が考えられる．DNS データベース上にはパスワードなどの非公開情報を書くことができず，また外部の DNS と接続していない企業ネットワーク内ではインターネット上のネームサーバにアクセスできないといった問題があるため，csd ではファイルを用いてボリューム情報を記述している．以下では，ボリューム情報を記述したファイルをボリュームファイルとよぶ．

ボリュームファイルに記述される情報としては，サーバの情報（ホスト名，パス名），代替サーバの情報，ボリュームの内容などがある．また，ボリュームに対するアクセス制御が必要な場合には，ユーザ名やパスワードなどのアクセス制御情報も記述することができる．

たとえば図5に示したボリュームファイルでは，CERT ボリュームの初期サーバとして大阪大学のサーバを用い，代替サーバとして cert.org のサーバを用いるということが記述されている．perl 言語で記述され

```
ftp-server:      ftp.center.osaka-u.ac.jp
ftp-directory:   CERT/advisories
ftp-server:      cert.org
ftp-directory:   pub/cert_advisories
description:     CERT Advisories
```

図5 ボリュームファイルの例
Fig. 5 An example of volume file.

た外部プログラムを利用すれば，これらのサーバを地理情報と障害発生頻度の順に並べ替えることができる．

代替サーバへの切替えは以下のようにして実現している．ボリュームごとに構造体を定義し，メンバとして代替サーバへのキュー（queue）を持つ．ボリュームへのアクセス要求が発生した場合はキューの先頭から順にサーバを検索し，障害が発生していないサーバを選ぶ．サーバはボリュームファイルに記述された順に並べられる．あるサーバへのコネクションが拒否されたり切断された場合，該当するサーバについて障害発生フラグを立て，次回のアクセス時に使用されないようにする．このとき，一定時間ののちにICMP echoを送り，echo replyが返ってきた場合はサーバの障害発生フラグを取り消すようにタイマとスレッドを設定する．

6. 評　価

ここではcsdの運用に基づいてアクセス装置を用いた場合の可用性の改善効果を評価する．1996年1月18日から1996年3月30日までの74日間にわたって奈良先端科学技術大学院大学においてcsdを運用した結果，2284本のFTP制御コネクションが張られ，ファイル転送のために5184本のFTPデータコネクションが張られた．FTP制御コネクションのうち119本は最大ユーザ数などの制限のためにFTP接続を拒否され，8本は接続先の計算機が応答しないためTCPコネクションを確立できなかった．FTPデータコネクションのうち42本は転送途中でデータが来なくなった．また，このほかにネームサーバが故障し，ホスト名からIPアドレスに変換することができず，接続先を特定できない場合が15回あった．これらの障害に対して，サーバ切替えを181回行い，障害を回避した．このうち，代替サーバを用意できない場合は合計5回であった．

以上から計算すると，$p_t = 5/2284 \leq 0.0022$ となり，奈良先端大におけるアクセス装置csdの可用性は99%以上であることが分かる．一方，サーバ切替えを行わなかった場合，$p_t = 186/2284$ となり，可用性は約92%であったと考えられる．このことから，アクセス装置を用いることで，アクセス装置を用いない場合と比べて可用性の改善効果が認められると考えられる．

7. 考　察

ボリュームを用いることで，サーバ切替えの対象となる資源を簡潔に記述することができる．また，アクセス装置においてボリュームごとにサーバ切替えを行うことで，障害発生時における可用性を改善することができる．前章で述べた評価結果から，このようなボリュームとアクセス装置を組み合わせた代替サーバの利用技術は有効であると考えられる．アクセス装置を用いず，クライアントにおいて代替サーバを選択した場合には，このような可用性の改善効果は得られないと考えられる．

クライアントにおいて代替サーバを選択する場合，可用性を改善するためには各々のサーバに関する詳細な知識をクライアントにおいて持つ必要がある．この手法では，クライアントにおいてサーバ選択アルゴリズムや障害検出のためのプロトコルを実装する必要が生じる．その結果，クライアントに対して高い機能が求められ，クライアントが複雑になるという問題点がある．また，クライアントにおいて代替サーバの可用性指標を持つことは困難であり，サーバ選択アルゴリズムを改善することによる可用性の改善は難しい．

本研究で提案した手法は，資源の複製が可能で，まとめて扱うことができることを前提としており，そのような特性を持つさまざまな資源においてアクセス装置およびボリュームを用いた可用性の改善効果が得られると考えられる．しかしながら，頻繁に内容が変更される資源に対してこの手法を適用した場合，資源の複製に要するトラヒックが増大するため，資源の重要度を勘案する必要がある．また，秘密にしなければならない情報を複製した場合，一般に資源の秘密性が低下し，セキュリティ的脅威に対する危険性が増す．このような危険は，セキュリティ機能を強化した代替サーバの構成手法を開発することによって軽減することができると考えられる．

8. 関連研究

代替サーバ機能を分散ファイルシステムにおいて実現したシステムとしては，Transarc社のAFS[9]，Auspex社のServerGuard[10]がある．OSF DCE/DFSの代替ファイルサーバ機能はAFSと同じものである．AFSでは，同一セル内部にしか代替ファイルサーバを置くことができず，広域ネットワーク上の異なるセルに代替ファイルサーバを置き，広域ネットワークにおけるサーバの可用性を改善することはで

きない．また，ServerGuardでは地理的に離れた複数のファイルサーバを相互にバックアップすることを主眼としており，ネットワークにおいて障害が発生したときにサーバを切り替える機構は用意されていない．

インターネットワーク上で代替ファイルサーバを構成する場合，一般に複製元のファイルサーバと複製ファイルサーバを区別し，ファイルサーバ間でファイル内容を同一に保つためのソフトウェアが実行される．このようなソフトウェアとしてmirrorやrdist[11]，sup[12]などがある．

インターネットにおける経路制御プロトコルの安定性については，興味深い結果が報告されている．Chinoy[13]は米国のバックボーンにおける経路情報の変動を観測し，参加組織の3%は12時間以内に10以上の経路変動を受けると報告している．また上水流[14]は日本のバックボーンを流れる経路情報を解析し，特定の接続形態において経路制御が不安定になることを指摘している．

より安定性の高い経路制御プロトコルについても研究が行われている．Perlman[15]は，誤った経路情報が流された場合においても，原因となるルータを取り外せば有限時間内に正常な状態に復帰するような経路制御プロトコルを提案している．

9. まとめ

本研究では，まずTCP/IPプロトコルスタックを用いたインターネットワークが次世代の情報通信基盤として十分な信頼性を提供していないことを指摘した．具体的には，トランスポート層における信頼性が低く，コネクション指向型トランスポートプロトコルにおいて約10%の確率でコネクションに障害が起きることを観測に基づいて示した．大多数のネットワークアプリケーションはトランスポートプロトコル上に直接実装されているため，ユーザに対して提供しているサービスの可用性が低い．

このような状況においてサービスの可用性を改善するため，前論文[3]においてアクセス装置を提案した．アクセス装置においてボリュームを単位としたサーバ切替えを行うことで，既存のクライアント・サーバ型アプリケーションに対して透過的に可用性の改善効果をもたらすことができると考えられる．しかしながら，従来の通信ソフトウェア実現方式では障害発生時に一時的にサーバが利用できなくなる等の問題点がある．

これを解決するため，本研究では従来とは異なる通信ソフトウェア実現方式を用いてアクセス装置csdを実装した．本方式では障害の早期発見，障害情報の共有，障害状態からの回復が可能であるという点で従来方式より優れていると考えられる．csdをインターネットにおいて運用して得られたデータに基づき，アクセス装置による可用性の改善効果を確認することができた．

謝辞 本研究を行うにあたり熱心に議論していただいたWIDEプロジェクトの皆様に感謝いたします．

参考文献

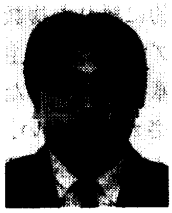
1) Clark, D.: The Design Philosophy of the DARPA Internet Protocols, *Proc. ACM SIGCOMM'88*, pp.106–114 (1988).
2) ISO7498/TC97/SC21: Information Processing Systems – Open Systems Interconnection – Basic Reference Model, ISO7498-1984 (1984).
3) 門林雄基，山口 英，宮原秀夫：Internetにおける資源アクセス装置の提案，情報処理学会論文誌，Vol.37, No.5 (1996).
4) Wright, G.R. and Stevens, W.R.: *TCP/IP Illustrated: The Implementation*, Vol.2, Addison-Wesley (1995).
5) Allman, E.: Sendmail – An Internetwork Mail Router, 4.4BSD System Manager's Manual, The USENIX Association, pp.SMM:9-1–SMM:9-12 (1994).
6) Mockapetris, P.V. and Dunlap, K.J.: Development of the Domain Name System, *Proc. ACM SIGCOMM'88*, pp.123–133 (1988).
7) Draves, R.P., Bershad, B.N., Rashid, R.F. and Dean, R.W.: Using Continuations to Implement Thread Management and Communication in Operating Systems, *Proc. 13th ACM Symposium on Operating System Principles*, pp.122–136 (1991).
8) Kadobayashi, Y., Yamaguchi, S. and Miyahara, H.: WWFS: An Evolutionary Framework for File Sharing in the Internet, *Proc. INET'93* (1993).
9) Transarc Corporation: AFS-3 Programmer's Reference.
10) Auspex Systems, Inc.: Auspex version 1.8M1 Software Release Note (1995).
11) Cooper, M.A.: Overhauling Rdist for the '90s, *LISA VI*, USENIX Association (1992).
12) Shafer, S. and Thompson, M.: The SUP Software Upgrade Protocol, Technical report, School of Computer Science, Carnegie Mellon University (1989).
13) Chinoy, B.: Dynamics of Internet Routing Information, *Proc. ACM SIGCOMM'93* (1993).
14) 上水流由香：マルチホーム接続のための経路制御について，情報処理学会研究報告，95-DPS-71-3

(1995).
15) Perlman, R.: Network Layer Protocols with Byzantine Robustness, Ph.D. Thesis, MIT (1988).

(平成8年 4 月15日受付)
(平成8年11月 7 日採録)

門林 雄基（正会員）

平成4年大阪大学基礎工学部情報工学科卒業．平成6年同大学大学院基礎工学研究科博士前期課程修了．平成8年同大学大学院博士後期課程を退学し，同大学大型計算機センター研究開発部助手．インターネットアーキテクチャ，分散ファイルシステムに関する研究に従事．電子情報通信学会，日本ソフトウェア科学会各会員．



山口 英（正会員）

昭和61年大阪大学基礎工学部情報工学科卒業．昭和63年同大学大学院博士前期課程修了．平成2年同大学大学院博士後期課程を退学し，同大学情報処理教育センター助手．以後，平成4年奈良先端科学技術大学院大学情報科学センター助手，同年同大学情報科学センター助教授を経て，平成5年同大学情報科学研究科助教授．工学博士（大阪大学）．インターネットアーキテクチャ，ネットワークセキュリティに関する研究に従事．電子情報通信学会，IEEE，Internet Society 各会員．

宮原 秀夫（正会員）

昭和42年大阪大学工学部通信工学科卒業．昭和47年同大学大学院博士課程修了．昭和48年京都大学工学部助手．昭和55年大阪大学基礎工学部助教授．昭和62年同大学大型計算機センター教授．平成元年同大学基礎工学部情報工学科教授．平成7年より同大学大型計算機センター長併任．昭和58～59年米国IBMトーマスワトソン研究所客員研究員．システム性能評価，マルチメディアシステム，広帯域通信網，ネットワーク管理に関する研究に従事．IEEE，電子情報通信学会各会員．